www.whooper.co.uk

This is the Personal Website of William Hooper
Email: email@whooper.co.uk
Tel: +44 78 38 18 98 07
Location: London, England

Home   **About**   **Photos**   **Articles**   **Miscellaneous**

...Asynchronous XLL functions and RTD...

<--Back to Excel Home Page

A conventional XLL worksheet function is synchronous. In other words Excel calls the function and waits until it returns a result. Users often need to add asynchronous functions. An example is a stock price which ticks at arbitrary times - a call back is needed which tells Excel "hey, new data has arrived please go and recalculate that cell in order to set it to the latest value". Another example is a long complex calculation which needs to be moved onto a separate thread.

Excel 2002 users should consider using the special RTD function for asynchronous calls (link). For earlier versions various approaches are possible. For market data the most popular is DDE server. Another possibility is a volatile function plus a calculation generating event. Although this approach works it is slow (see this link). A function with a dummy global refresh dependency argument that is dirtied by a refresh event is better but, as with a volatile function, all market data, and all downstream dependent cells, tick on any update. A function with a topic unique refresh dependency is very efficient. An xla contains a range name for each topic, and the topic value can be stored and easily updated inside this name. Also, the worksheet function can call automatic formula editing code to ensure that the dependency argument passed to it is correct. This is my recommended solution for asynchronous addins in Excel 97-2000. Another interesting possibility, less efficient but very impressive looking, is to track cell locations inside an automated XLL worksheet function by finding and holding the cells IDispatch range pointer, then calling an XLL update macro to find and dirty the formulas with .formula=.formula (Unfortunately you can not use range.calculate because Excel does not mark a cell manually calculated in this way as dirty, therefore downstream dependent cells will not refresh. Also, for completeness, it should be said that using the XLL hidden name space (which can be set even inside an xll worksheet function) to track more than a couple of hundred cells is too slow).

You can experiment with cell tracking concept in vba:

```
Option Explicit

Dim r As Range
Dim v As Double

Function GetVal()
Set r = Application.ThisCell
```

```
GetVal = v
End Function

Sub Update()
v = Rnd()
r.Formula = r.Formula
End Sub
```

If you drag and drop the worksheet function cell around the page you will notice that the update macro continues to track it and is able to refresh it. The tracking fails when the function is moved to another page; but that's not so bad because when the function is moved to anther page it will be marked as dirty and will recalculate, thus giving the function a chance to refresh the link.